Check 31. 2018 Scopus - Full - Software size measurement of student information terminal with use case point.pdf

by Arta Sundjaja

Submission date: 27-May-2019 10:14AM (UTC+0700)
Submission ID: 1102517865
File name: surement_of_student_information_terminal_with_use_case_point.pdf (906.44K)
Word count: 6038
Character count: 32078

Reprinted From

ISBN: 978-153860783-1

11

2017 IEEE International Conference on Cybernetics and Computational Intelligence, CyberneticsCOM 2017 - Proceedings

Volume 2017-November, 9 March 2018, Pages 164-169
http://ieeexplore.ieee.org

SOFTWARE SIZE MEASUREMENT OF STUDENT INFORMATION TERMINAL WITH USE CASE POINT

[1] Dede Kurniadi, [2]Sasmoko, [3]Harco Leslie Hendric Spits Warnars, [4]Ford Lumban Gaol2

[1]Computer Science Department, BINUS Graduate Program – Doctor of Computer Science Bina Nusantara University Jakarta, Indonesia 11480
[2] Primary Teacher Education Department, Faculty of Humanities and Research Interest Group in Educational Technology, Bina Nusantara University, Jakarta, Indonesia, 11480
[3-4]Computer Science Department, BINUS Graduate Program – Doctor of Computer Science Bina Nusantara University Jakarta, Indonesia 11480

URL: *http://ieeexplore.ieee.org/document/8311703*

Scopus

Document details

2017 IEEE International Conference on Cybernetics and Computational Intelligence, CyberneticsCOM 2017 - Proceedings
Volume 2017-November, 9 March 2018, Pages 164-169
2017 IEEE International Conference on Cybernetics and Computational Intelligence, CyberneticsCOM 2017; KEE Resort and Spa PatongPhuket; Thailand; 20 November 2017 through 22 November 2017; Category numberCFP1730S-ART; Code 135216

Software size measurement of student information terminal with use case point (Conference Paper)

Kurniadi, D.[a], Sasmoko, S.[b], Warnars, H.L.H.S.[a], Gaol, F.L.[a]

[a]Computer Science Department, BINUS Graduate Program, Bina Nusantara University, Jakarta, 11480, Indonesia
[b]Primary Teacher Education, Faculty of Humanities, Bina Nusantara University, Jakarta, 11480, Indonesia

Abstract

Student Information Terminals (S-IT) is an independent academic service information system for students, where this service makes it easy for students to obtain academic information in real time with information such as the transcript of academic achievement, finance, course, attendance, exam, lecturer, card examinees and announcements academic, and has the function to print directly the data independently on S-IT devices. To find out how well the S-IT is in terms of software size, then needed a measurement. The measurements used in this paper using the Use Case Point (UCP) method as one of the approved software metrics which measure the functionality our software size. The results of the measurement of software size S-IT shown that the project has a small size, the software has a value of UCP = 96.767 estimate effort, has the development time 1,452 hours or equivalent 9months 1 week and have development costs in

View references (22)

Cited by 2 documents

Design of decision support system for blood analysis
Septiana, Y. , Kurniadi, D. , Mulyani, A.
(2018) MATEC Web of Conferences

Design of prototype decision support system for flood detection based on ultrasonic sensor

Indonesian Rupiah is 263,175,000 IDR. The aims of measurement software size S-IT with the use case point is to help make decisions about the implementation of software development application project in terms of the estimated time, costs, and people. © 2017 IEEE.

SciVal Topic Prominence

Topic: Students | Radio frequency identification (RFID) | student attendance

Prominence percentile: 80.773

Author keywords

information system academic | software measurement | software metrics | student information terminal | use case point

Indexed keywords

Engineering controlled terms: Application programs | Artificial intelligence | Cost benefit analysis | Cost estimating | Information systems | Software design | Students | Teaching

Engineering uncontrolled terms: Academic achievements | Development costs | Information terminal | Measurement software | Service information | Software Measurement | Software metrics | Use case points

Engineering main heading: Information use

ISBN: 978-153860783-1
Source Type: Conference Proceeding
Original language: English
DOI: 10.1109/CYBERNETICSCOM.2017.8311703
Document Type: Conference Paper
Sponsors: IEEE Indonesia
Publisher: Institute of Electrical and Electronics Engineers Inc.

Related documents

Software size measurement with use case point for employee application software at STT-PLN

Agtriadi, H.B. , Chandra, N. , Warnars, H.L.H.S.
(2018) 2017 IEEE International Conference on Cybernetics and Computational Intelligence, CyberneticsCOM 2017 - Proceedings

Use case point as software size measurement with study case of Academic Information System

Safrizal, S. , Warnars, H.L.H.S. , Gaol, F.L.
(2018) 2017 IEEE International Conference on Cybernetics and Computational Intelligence, CyberneticsCOM 2017 - Proceedings

Quality measurement for serious games

Suryapranata, L.K.P. , Gaol, F.L. , Soewito, B.

References (22)

View in search results format >

All | Export | Print | E-mail | Save to PDF | Create bibliography

1 Mulyani, A., Kurniadi, D.
Analisis penerimaan teknologi student information terminal (s-IT) Dengan Menggunakan Technology Acceptance Model (TAM)
(2015) *Jurnal Wawasan Ilmiah*, 7 (12), pp. 23-35.

2 Warnars, H.L.H.S., Randriatoamanana, R.
Datawarehouser: A Data Warehouse artist who have ability to understand data warehouse schema pictures
(2016) *IEEE TENCON 2016 (Technologies for Smart Nation)*, pp. 2207-2210. Cited 6 times.
22-25 Nov Singapore

3 Warnars, H.L.H.S.
Perbandingan penggunaan Database OLTP (Online Transactional Processing) dan Data Warehouse
(2014) *Creative Communication and Innovative Technology (CCIT) Journal*, 8 (1), pp. 83-100. Cited 9 times.
September

4 Warnars, H.L.H.S.
Mining patterns with attribute oriented induction
(2015) *The Int. Conf. on Database, Data Warehouse, Data Mining and Big Data (DDDMBD2015)*, pp. 11-21. Cited 7 times.
Tangerang, Indonesia 10-12 September 2015

(2017) Proceedings - 2017 International Conference on Applied Computer and Communication Technologies, ComCom 2017

View all related documents based on references

Find more related documents in Scopus based on:

Authors > Keywords >

5 Warnars, H.L.H.S., Wijaya, M.I., Tjung, H.B., Xaverius, D.F., Van Hauten, D., Sasmoko

Easy understanding of Attribute Oriented Induction (AOI) characteristic rule algorithm

(2016) *International Journal of Applied Engineering Research*, 11 (8), pp. 5369-5375. Cited 7 times.
http://www.ripublication.com/ijaer.htm

6 Warnars, H.L.H.S.

Using attribute oriented induction high level emerging pattern (AOI-HEP) to mine frequent patterns (Open Access)

(2016) *International Journal of Electrical and Computer Engineering*, 6 (6), pp. 3037-3046. Cited 11 times.
http://iaescore.com/journals/index.php/IJECE/article/view/5921/5583
doi: 10.11591/ijece.v6i6.10579

View at Publisher

7 Warnars, H.L.H.S.

Mining frequent and similar patterns with attribute oriented induction high level emerging Pattern (AOI-HEP) Data Mining Technique

(2014) *International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)*, 3 (11), pp. 266-276. Cited 14 times.
EISSN : 2279-0047

8 Warnars, S.

Mining frequent pattern with Attribute Oriented Induction High Level Emerging Pattern (AOI-HEP)

(2014) *2014 2nd International Conference on Information and Communication Technology, ICoICT 2014*, art. no. 6914056, pp. 149-154. Cited 12 times.
ISBN: 978-147993581-9
doi: 10.1109/ICoICT.2014.6914056

View at Publisher

9 Warnars, S.
Attribute Oriented Induction High Level Emerging Pattern (AOI-HEP) future research
(2014) *Proceedings of 2014 International Conference on Information, Communication Technology and System, ICTS 2014*, art. no. 7010470, pp. 13-18. Cited 7 times.
ISBN: 978-147996857-2
doi: 10.1109/ICTS.2014.7010470
View at Publisher

10 Iskandar, K., Gaol, F.L., Soewito, B., Warnars, H.L.H.S., Kosala, R.
Software size measurement of knowledge management portal with use case point
(2016) *Proceeding - 2016 International Conference on Computer, Control, Informatics and its Applications: Recent Progress in Computer, Control, and Informatics for Data Science, IC3INA 2016*, art. no. 7863021, pp. 42-47. Cited 5 times.
ISBN: 978-150902323-3
doi: 10.1109/IC3INA.2016.7863021
View at Publisher

11 Karner, G.
Resource estimation for objectory projects
(1993) *Objective Systems. SF AB*. Cited 42 times.

12 Dewi, R.S., Subriadi, A.P.
Use case point-activity-based costing: Metode baru untuk mengestimasi biaya pengembangan perangkat lunak
(2015) *SISFO*, 5.

13 Fenton, N., Bieman, J.
Software Metrics: A Rigorous and Practical Approach, 2014. Cited 1809 times.
CRC Press

14 Nassif, A.B., Capretz, L.F., Ho, D.
Calibrating use case points

(2014) *36th International Conference on Software Engineering, ICSE Companion 2014 - Proceedings*, pp. 612-613. Cited 11 times.
ISBN: 978-145032768-8
doi: 10.1145/2591062.2591141

View at Publisher

15 Alves, L.M., Sousa, A., Ribeiro, P., Machado, R.J.
An empirical study on the estimation of software development effort with use case points

(2013) *Proceedings - Frontiers in Education Conference, FIE*, art. no. 6684796, pp. 101-107. Cited 7 times.
ISBN: 978-146735261-1
doi: 10.1109/FIE.2013.6684796

View at Publisher

16 Clemmons, R.K.
Project estimation with Use Case Points

(2006) *CrossTalk*, 19 (2), pp. 18-22. Cited 28 times.
http://www.stsc.hill.af.mil/crosstalk/2006/02/0602Clemmons.pdf

17 Nageswaran, S.
Test effort estimation using use case points
(2001) *Quality Week*, pp. 1-6. Cited 19 times.

18 Anda, B.
Comparing effort estimates based on use case points with expert estimates
(2002) *Empirical Assessment in Software Engineering (EASE 2002), Keele, UK*. Cited 10 times.

19 Carroll, E.R.
Estimating software based on use case points
(2005) *Companion to the 20th Annual ACM SIGPLAN Conference on Object-oriented Programming, Systems, Languages, Applications*, pp. 257-265. Cited 33 times.

20 Kurniawan, W., Sholiq, S., Sutanto, T.
Penentuan effort rate pada estimasi effort menggunakan metode use case point untuk pengembangan perangkat lunak website kepemerintahan
(2013) *Jurnal JSIKA*, 2 (2), pp. 61-71.

21 Kurniadi, D.
Perancangan arsitektur sistem e-academic dengan konsep kampus digital menggunakan unified software development process (USDP)
(2014) *Jurnal Wawasan Ilmiah*, 5 (10), pp. 1-16.

22 Kurniadi, D., Mulyani, A.
Implementasi pengembangan student information terminal (S-IT) Untuk Pelayanan Akademik Mahasiswa
(2016) *Jurnal Algoritma*, 13 (1), pp. 437-442.

Software Size Measurement of Student Information Terminal with Use Case Point

Dede Kurniadi,
Computer Science Department,
BINUS Graduate Program – Doctor of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
dede.kurniadi@binus.ac.id

Sasmoko,
Primary Teacher Education Department,
Faculty of Humanities
Bina Nusantara University
Jakarta, Indonesia 11480
sasmoko@binus.edu

Harco Leslie Hendric Spits Warnars[1],
Ford Lumban Gaol[2]
Computer Science Department, BINUS Graduate Program – Doctor of Computer Science
Bina Nusantara University
Jakarta, Indonesia 11480
Spits.shendric@binus.ac.id[1],
fgaol@binus.edu[2]

***Abstract*—Student Information Terminals (S-IT) is an independent academic service information system for students, where this service makes it easy for students to obtain academic information in real time with information such as the transcript of academic achievement, finance, course, attendance, exam, lecturer, card examinees and announcements academic, and has the function to print directly the data independently on S-IT devices. To find out how well the S-IT is in terms of software size, then needed a measurement. The measurements used in this paper using the Use Case Point (UCP) method as one of the approved software metrics which measure the functionality our software size. The results of the measurement of software size S-IT shown that the project has a small size, the software has a value of UCP 96.767 estimate effort, has the development time 1,452 hours or equivalent 9 months 2 week and have development costs in Indonesian Rupiahs 263,175,000 IDR. The aims of measurement software size S-IT with the use case point is to help make decisions about the implementation of software development application project in terms of the estimated time, costs, and people.**

Keywords—use case point; software measurement; software metrics; student information terminal; information system academic

I. INTRODUCTION

Student Information Terminals (S-IT) is a software system for academic student services information-based touch screen and is a self-contained platform service for students that provide the ease of use of information systems and academic services to students. S-IT is used the first time in 2013 and is currently used by three existing colleges in Garut, West Java, Indonesia, such as AMIK Garut, Garut University (the Faculty of engineering, pharmacy, political science), and School of Technology Garut (STTG) and feature continue to be developed until recently. Development of S-IT was built using RAD Studio, with the intention to complement the academic campus information systems that already exist, but the only use of it is devoted to the student with the retrieval of the data directly from the database of the academic center-based Oracle.

Development and utilization of S-IT is a means of support for achieving the objectives of the Organization, it can be done effectively if members of an organization have the capability and expertise in using and developing S-IT well, for it to be done efforts to find out the effectiveness and efficiency in the development and utilization of Student Information Terminal. The utilization of previous technology acceptance ever done research on Acceptance Technology of Student Information Terminal using Technology Acceptance Model (TAM) method, with the external variables are tested include the design of the portal, e-resources, organization and user abilities and skill, with the conclusion of the results of hypothesis testing, is accepted [1]. This S-IT application will be extended its intelligence with Data Warehouse [2], [3] or/and data mining technologies and can use data mining algorithm such as using Attribute Oriented Induction technique (AOI) [4], [5] or AOI-HEP in order to find frequent or similar patterns [6], [7], [8], [9].

Along with the improvement of the development features of S-IT, need further measurements in terms of the size of the software S-IT, where the size of the software be measured based on the length, complexity, functionality, and reuse, one of the methods of measurement used to measure software is to use the method of the Use Case Point (UCP) approach. A similar study on measurements using use case points had also been done by Parto et al, regarding the Software Size Measurement of Knowledge Management Portal with Use Case Points [10].

Use Case Point (UCP) is a method which has the ability to provide an estimate of the effort necessary to make a project based on the number and complexity of the use-case that is owned by the software project [11]. The advantages of the method of use case points i.e. can provide a closer estimate actual estimate resulting from the manufacturing or software development experience [12]. Use case point is optional as approved software metrics which can measure functionality our software size [13], [14]. UCP measure our software size, the complexity and count effort to know how many man-days to develop this S-IT [13], [15], [16], as for further explanation concerning about the Use Case Points will be described in part 2. In this paper, we will calculate the size of Student Information Software Terminal method using Use Case points.

II. METHOD MEASUREMENT EXPLANATION

A. Use Case Point Method

The method of UCP is a method which has the ability to provide an estimate of the effort necessary to make a project relied on the number and the number of complexity of use case, owned by the software project [11]. According to another opinion, the UCP is a method that can analyze the actor, use

case, and a variety of technical factors and environmental factors to be an equation [16]. The advantages of the method of use case points i.e. can provide a closer estimate actual estimate resulting from the manufacturing or software development experience. This is evidenced by several studies ever done before, and produce the statement that, UCP has 6% deviation [17], UCP has a deviation of 19%, while experts estimate has a deviation of 20% [18], UCP has a deviation of 9% [19].

The method of UCP is very important to do the estimation effort, where the value of effort estimation obtained from the results of the multiplication between UCP and value Effort Rate (ER). The value of the Effort Rate (ER) is the value of the variable used to calculate effort [20].

B. *Step by step calculation Use Case Point*

Step by step process to perform the calculation of the use case point is done in accordance with the formulated by Karner [11], which are divided into 3 stages as follows:

1. Calculate Unadjusted Use Case Point (UUCP)

   The first step at this stage is to determine Unadjusted Actor Weights (UAW) as a simple, average, or complex according the table I as follows:

TABLE I. WEIGHTED ACTOR

Complexity	*Weight*	*Description*
Simple	1	Define with API
Average	2	Interaction via protocol TCP/IP
Complex	3	Interaction with GUI or Web Page

   Total Unadjusted Actor Weights (UAW) derived from counting the number of actors based on each type (level of complexity), and multiply with total weight for each factor corresponds to the table so that the obtained formulas.

$$UAW = \sum (\#Actors * Weight\ Factor) \quad (1)$$

   Then the second step at this stage is to determine Unadjusted Use Case Weights (UUCW), how to calculate UUCW the same as calculating the UAW, that each use case is split into three groups, the simple or average or complex, depending on the number of transactions carried out. For a more detailed explanation of descriptions use case can be seen in the following table II:

TABLE II. WEIGHTED USE CASES

Complexity	*Weight*	*Description*
Simple	1	Using transaction <= 3
Average	2	Using transaction 4 to 7
Complex	3	Using transaction > 7

   Total Unadjusted Use Case Weights (UUCW) derived from counting the number of use-case from each level of complexity multiplied by total factor of each use case. Formula to calculate the UUCW are as follows

$$UUCW = \sum (\#Use\ Cases * Weight\ Factor) \quad (2)$$

   Then total the UAW and UUCW to get the Unadjusted Use Case Point (UUCP), as in the following formula:

$$UUCP = UAW + UUCW \quad (3)$$

2. Calculate Use Case Point (UCP)

   In the calculation of the value of the Use Case Point (UCP) contained the value of complexity factor. The notion of complexity factors are factors that influence directly in the process of the working of the software project. Complexity factor is split into two groups, namely:

   - Technical Complexity Factor (TCF).
   - Environmental Factor (EF)

   Here's an explanation of each of the complexity factors are presented in the following table III:

TABLE III. TECHNICAL COMPLEXITY FACTOR (TCF)

	Technical Factor	*Weight*
1	Distributed System Required	2
2	Response Time is Important	1
3	End-User Efficiency	1
4	Complex Internal Processing Required	1
5	Reusable Code Must Be A Focus	1
6	Installation easy	0.5
7	Usability	0.5
8	Cross-platform support (Portability)	2
9	Easy to change	1
10	Highly concurrent	1
11	Custom security	1
12	Dependence on third-part code	1
13	User training	1

   The values of TCF is multiplied by a weighting value to each one. The weighting values from 0 to 5 numbers given to each factor depends on how big the influence of these factors. 0 means do not affect, 3 means the average, and 5 means giving a great influence. The results of the multiplication of the values and weights are then summed up to obtain total of Technical Factor (TF), and then will be used to have Technical Complexity Factor (TCF).

$$TCF = 0.6 + (0.01 * TFactor) \quad (4)$$

   The values on the environmental factor multiplied by a weighting value to each one. Weighting the given value from 0 to 5 on each factor depends on how big the influence of these factors. 0 means do not affect, 3 means the average, and 5 means giving a great influence. The results of the

multiplication of the values and weights are then summed up to get the total EF.

$$EF = 1.4 + (-0.03 * EFactor) \quad (5)$$

TABLE IV. ENVIRONMENTAL FACTOR (EF)

	Environmental Factor	*Weight*
1	Familiarity with the Project	1.5
2	Application Experience	0.5
3	Programming Experience	1
4	Analyst Capability	0.5
5	Motivation	1
6	Stable Requirements	2
7	Part Time Staff	-1
8	Difficult Programming Language	-1

Finally, we can get the value of the Use Case Point (UCP) through the UUCP multiply with TCF and EF.

$$UCP = UUCP * TCF * EF \quad (6)$$

3. Calculate Estimate Effort (E) in person-hours.

At the end, after calculating the above variables shown in the previous stage, the last step is to calculate the E as effective effort which count person-hours (PH) which multiply UCP with number of PH per UCP (PHperUCP) as shown in formula (7).

$$E = UCP * PHperUCP \quad (7)$$

Effort rate which is used is the ratio of hour number per use case point based on past projects. If suppose, the project is a new project and there is no history data that were collected, then the used value that ranges from 15 to 30. However, the most commonly used value is a number 20 [16].

III. STUDENT INFORMATION TERMINAL CASE STUDY

Student Information Terminals (S-IT) is an independent platform for student service, this service makes it easy for students to obtain academic information in real-time starts from transcript of academic achievement, financial, schedule of lectures, attendance, exam, lecturer, cards and announcements academic, students also can print directly the data independently or sent via email as an academic archive, as well as other digital information such as job vacancies and other information connected with the central database of the academic campus.

Based on the features and services of S-IT was available, then to measure the size of the software S-IT is by using a Use-Case in Point. As for the use case S-IT was first developed on the basis of the implementation of the design of the system architecture of e-academic with the concept of digital campus [21], [22] the following use case diagrams of software S-IT can be seen in figure 1.

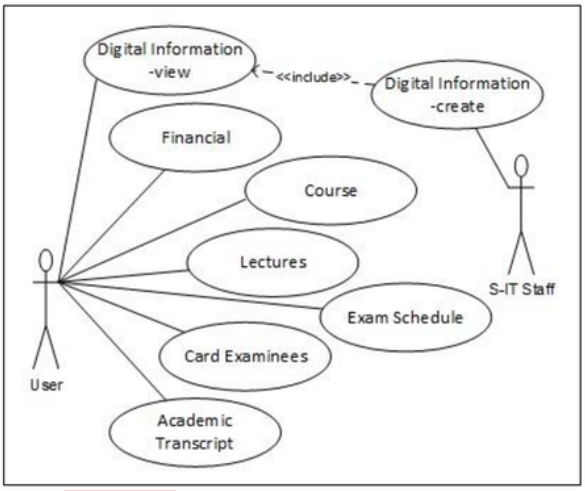

Fig. 1. Use Case Diagram Student Information Terminal

As shown in figure 1 use case diagram of the S-IT consists of two actors, namely students, S-IT Staff and 8 activity use cases, such as:

1. Use case Digital Information View, providing up-to-date information to students such as, announcements of academic, scholarship information, and job vacancies.
2. Use case Financial, provide information in detail about the history of bills and payments.
3. Use case Course, provide information regarding the taking of courses taken students and courses approved by the academic supervisor.
4. Use case Lectures, provide information in detail about the schedule of lectures, a lecturer, and the room.
5. Use case Exam Schedule, providing information on the schedule of midterm and final exam of the semester.
6. Use case Card Examinees, printing services card test participants include middle test and the final exam of the semester.
7. Academic Transcript Use case, where this use case provides information about the temporary transcript of academic achievement based on studies that have been taken by students.
8. Digital Information Create Use case, where this use case tells about the create Digital Information by S-IT staff.

The 1st stage in UCP is Calculate UUCP by adding UAW and UUCW. The following identification and calculation process for UAW presented in table V and UUCW in table VI.

Actors in figure 1 are categorized as a complex actor since all 2 actors in figure 1 are person who interacts by GUI and automatically each was weighted with 3. Table V shows UAW = 6 as result of formula (1) where is summarization of weight score and number of actors.

TABLE V. IDENTIFY UNADJUSTED ACTOR WEIGHT (UAW)

Category	*Weight*	*Actors*	*Count*	*Weight * Count*
Simple	1	-	0	0
Average	2	-	0	0
Complex	3	User, S-IT Staff	2	6

The 8 use cases which shows in figure 1 are categorized become simple or average or complex. There are 3 use cases categorizes as simple such as Digital Information, Lectures, Exam Schedule use cases. Next, there are 2 average category use cases such as Course, Card Examinees, and 3 complex use cases such as Financial, Academic Transcript, Digital Information Create. Moreover, table VI shows the result of categorization of use cases in figure 1, which categorized with 5 for simple, 10 for average and 15 for complex. UUCW = 80 in table VI is a result formula (2) where summarisation of multiplication weighted score with each use case category.

TABLE VI. IDENTIFY UNADJUSTED USE CASE WEIGHT (UUCW)

Use Case	*Category*	*Weight*
Digital Information View	Simple	5
Financial	Complex	15
Course	Average	10
Lectures	Simple	5
Exam Schedule	Simple	5
Card Examinees	Average	10
Academic Transcript	Complex	15
Digital Information Create	Complex	15
Total Unadjusted Use Case Weight (UUCW)		**80**

The score UUCP=86 based on formula (3) where it be done based on addition of UAW and UUCW where UAW=6 as shown in table V and UUCW=80 as shown in table VI.

$$\begin{aligned} UUCP &= UAW + UUCW \\ &= 6 + 80 \\ &= 86 \end{aligned}$$

The second stage UCP is calculation of UCP with calculation TCF score from formula (4) and EF score from formula (5) as shown in Tables VII and VIII respectively. The number of TCF is 0.97 from formula (4) where score TFactor is 37 as summarization of factors in table VII,

$$\begin{aligned} TCF &= 0.6 + (0.01 * TFactor) \\ &= 0.6 + (0.01 * 37) \\ &= 0.97 \end{aligned}$$

The number of EF is 1.16 from formula (5) where Environmental Factor is 8 as summarization of factors in table VIII,

$$\begin{aligned} EF &= 1.4 + (-0.03 * EFactor) \\ &= 1.4 + (-0.03 * 8) \\ &= 1.16 \end{aligned}$$

TABLE VII. CALCULATE TECHNICAL COMPLEXITY FACTOR (TCF)

	Technical Factor	*Weight*	*Value*	*Weight * Value*
1	Distributed System Required	2	3	6
2	Response Time is Important	1	5	5
3	End User Efficiency	1	5	5
4	Complex Internal Processing Required	1	1	1
5	Reusable Code Must Be A Focus	1	3	3
6	Installation easy	0.5	4	2
7	Usability	0.5	4	2
8	Cross-platform support (Portability)	2	1	2
9	Easy to change	1	4	4
10	Highly concurrent	1	3	3
11	Custom security	1	3	3
12	Dependence on third-part code	1	0	0
13	User training	1	1	1
Total Technical Factor (TFactor)				**37**

TABLE VIII. CALCULATE ENVIRONMENTAL FACTOR (EF)

	Environmental Factor	*Weight*	*Value*	*Weight * Value*
1	Familiarity with the Project	1.5	2	3
2	Application Experience	0.5	2	1
3	Programming Experience	1	3	3
4	Analyst Capability	0.5	4	2
5	Motivation	1	3	3
6	Stable Requirements	2	3	6
7	Part Time Staff	-1	5	-5
8	Difficult Programming Language	-1	5	-5
Total Environmental Factor (EFactor)				**8**

Having obtained the value of TCF as shown with formula (4) and EF values with formula (5), next is calculation process UCP by formula (6), where the number of UCP is 96.767 based on the following calculation:

$$\begin{aligned} UCP &= UUCP * TCF * EF \\ &= 86 * 0.97 * 1.16 \\ &= 96.767 \end{aligned}$$

The third stage or final stage in Use-Case is calculated the effective effort as E in person-hours as PH by multiplication of UCP and specific value for Person Hour for each UCP (PHperUCP) based on formula (7). S-IT development projects including the category of simple/low, these categories are based on table IX [10].

TABLE IX. SOFTWARE COMPLEXITY PHPERUCP CATEGORIES

Category	*Person-Hour per UCP PHperUCP*
Simple/Low	1 – 20
Complex	21 – 40
Very Complex/High	> 41

In determining PHPperUCP figures based on the results of the interview with a project leader who developed the software S-IT, obtained numbers PHperUCP = 15 hours for the development of Students Information Terminal. The following details the calculation of Effort Estimation:

$$E = UCP * PHperUCP$$

$$= 96.767 * 15$$

$$= 1,451.505 \text{ hours}$$

Using formula (7) then E score as Estimation Effort is created by multiplication of UCP= 96.767 with 15 and E score is 1,451.505 hours and rounded to 1,452 hours.

IV. RESULT AND ANALYSIS

In section result and analysis, we will analysis the result calculation software metric which was based on four categories of software size shown in table X [10]. Based on table X there are four categories of software size they are such as Small or Medium or Large or Extreme. Where small has less than 99 UCP, medium has UCP between 100 to 299, medium has UCP between 300 to 799 and extreme has UCP more than 799. Using formula (6) then UCP score is 96.767 which categorized with table X as small category. Thus, software size for S-IT is categorized as Small project.

TABLE X. SOFTWARE SIZE CATEGORIES

Category	*Use Case Point*
Small	< 99
Medium	100 -299
Large	300 – 799
Extreme	>= 800

But based on interviews with the project leader in doing this S-IT project, the average working time per person following the regulation time in general i.e. 5-day work week with a time of 8 hours for each day. Moreover, based on the results of the formula (7) as E for to estimate the development of S-IT was 1,452, so if the conclusion is drawn to work on the project S-IT is a maximized 1,452 hours. If based on the hours of work 8 hours per day, then an effort estimate this time divided by 8 hours so having time 181.5 day (1,452/8 = 181.5). In addition, 258 days divided by 5 as a 5-day work week and it has 22.6 weeks (181.5/5 = 22.6). Furthermore, 22.6 weeks were divided by 4 where will have 4 weeks per month and its means has a 9.1 month (36.3/4 = 9.1). Thus, the time required to develop the project S-IT is a maximum of 9 months (9 months 1 week). If calculated in terms of financing based on information obtained from the project leader, States that every worker IT has varying per month salary adjusted to his craft, the average for man-days is 1,450 million IDR. Appropriate Regional Minimum Wages. Finally, the project S-IT has the value of the project is around 263,175,000 IDR, this figure is obtained based on the calculation of the salary man-day in Indonesian Rupiah is 1,450,000 IDR multiplied with 181.5 days (181.5 days * 1,450,000 IDR= 263,175,000 IDR).

TABLE XI. SUMMARY OF S-IT MEASUREMENT USING USE CASE POINT

Description	*S-IT*
Unadjusted Actor Weight (UAW)	6
Unadjusted Use Case Weight (UUCW)	80
Unadjusted Use Case Points (UUCP)	86
Technical Complexity Factor (TCF)	0.97
Environmental Factor (EF)	1.16
(Adjusted) Use Case Points (UCP)	96.767
Software Size	< = 99 (Small)
Software Complexity PHperUCP	1 – 20 (15) (Simple)
(Simple) Estimate effort	1,452 hours
Project Value	263,175,000 IDR

V. CONCLUSIONS

The measurement of the size of the software with Use Case Points can help a developer or management to make better decisions in terms of how to handle the project, how long to grow and how much to invest. In addition, measurement using use case point estimation of effort, the determination method of the time, the value of the project, the number of people involved in this project can also be measured. The measurement of the size of the software with the software metrics such as Use Case Point measurement is based on the measuring of the internal product attributes can be measured simply by virtue of the entity itself. In addition, the Use Case Point part of the measurement of the size of the software in terms of function where the provided function measured by product to the user.

Measurement of Software size can reflect the effort, cost, and productivity. The measurement of the size of the software by using the Use Case Point is very useful where the size of the software can be measured based on use case diagrams. In this case use of a chart paper cases for the project S-IT, measurements show that the project has the size of a small piece of software with a Use Case Point (UCP) = 96.767 and have a business estimation will be developed at 1,452 hours and have construction costs amounting to 263,175,000 IDR as shown in table XI.

REFERENCES

[1] A. Mulyani and D. Kurniadi, "Analisis Penerimaan Teknologi Student Information Terminal (S-IT) Dengan Menggunakan Technology Acceptance Model (TAM)," *Jurnal Wawasan Ilmiah.*, vol. 7, no. 12, pp. 23–35, 2015.

[2] H. L. H. S. Warnars, R. Randriatoamanana (2016), "Datawarehouser: A Data Warehouse artist who have ability to understand data warehouse schema pictures", IEEE TENCON 2016 (Technologies for Smart Nation), pp. 2207-2210, 22-25 Nov 2016, Singapore.

[3] H. L. H. S. Warnars, (2014), "Perbandingan penggunaan Database OLTP (Online Transactional Processing) dan Data Warehouse", Creative Communication and Innovative Technology (CCIT) journal, Vol. 8 No. 1, pp. 83-100, September 2014. ISSN: 1978-8282.

[4] H. L. H. S. Warnars (2015), "Mining Patterns with Attribute Oriented Induction", The Int. Conf. on Database, Data Warehouse, Data Mining and Big Data (DDDMBD2015), Tangerang, Indonesia, pp. 11-21, 10-12 September 2015.

[5] H. L. H. S. Warnars, M. I. Wijaya, H. B. Tjung, D. F. Xerius, D. V. Hauten, Sasmoko (2016), "Easy understanding of Attribute Oriented Induction (AOI) characteristic rule algorithm", International journal of Applied Engineering Research (IJAER), vol. 11, No. 8, pp. 5369-5375. ISSN-0973-4562.

[6] H. L. H. S. Warnars, "Using Attribute Oriented Induction High Level Emerging Pattern (AOI-HEP) to Mine Frequent Patterns," International Journal of Electrical and Computer Engineering (IJECE), vol. 6, no. 6, pp. 3037–3046, 2016..

[7] H. L. H. S. Warnars, (2014), "Mining Frequent and Similar Patterns with Attribute Oriented Induction High Level Emerging Pattern (AOI-HEP) Data Mining Technique", International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS), vol. 3, Issue 11, pp.266-276. EISSN : 2279-0047, ISSN: 2279-0055.

[8] H. L. H. S. Warnars (2014),"Mining Frequent Pattern with Attribute Oriented Induction High level Emerging Pattern (AOI-HEP)", IEEE the 2nd Int. Conf. on Information and Communication Technology (IEEE ICoICT 2014), Bandung, Indonesia, pp. 144-149, 28-30 May 2014.

[9] H. L. H. S. Warnars (2014), "Attribute Oriented Induction High Level Emerging Pattern (AOI-HEP) future research", IEEE the 8th Int. Conf. on Information & Communication Technology and Systems Systems (ICTS), Surabaya, Indonesia, pp. 13-18, 24-25 September 2014.

[10] K. Iskandar, F. L. Gaol, B. Soewito and H.L.H.S. Warnars (2016), "Software size measurement of knowledge management portal with use case point", The Int. Conf. on Computer, Control, Informatics, and its Applications (IC3INA 2016), 3-5 Oct 2016, Tangerang, Indonesia.

[11] G. Karner, "Resource estimation for objectory projects," *Objective Systems. SF AB*, 1993.

[12] R. S. Dewi, A. P. Subriadi, and others, "Use Case Point-Activity-Based Costing: Metode Baru untuk Mengestimasi Biaya Pengembangan Perangkat Lunak," *SISFO*, vol. 5, 2015.

[13] N. Fenton and J. Bieman, *Software metrics: a rigorous and practical approach*. CRC Press, 2014.

[14] A. B. Nassif, L. F. Capretz, and D. Ho, "Calibrating use case points," in *Companion Proceedings of the 36th International Conference on Software Engineering*, 2014, pp. 612–613.

[15] L. M. Alves, A. Sousa, P. Ribeiro, and R. J. Machado, "An empirical study on the estimation of software development effort with use case points," in *Frontiers in Education Conference, 2013 IEEE*, 2013, pp. 101–107.

[16] R. K. Clemmons, "Project Estimation With Use Case Points," *Journal of Defense Software Engineering.*, pp. 18–22, 2006.

[17] S. Nageswaran, "Test Effort Estimation Using Use Case Points," in *Quality Week*, 2001, pp. 1–6.

[18] B. Anda, "Comparing effort estimates based on use case points with expert estimates," *Empirical Assessment in Software Engineering (EASE 2002), Keele, UK*, 2002.

[19] E. R. Carroll, "Estimating software based on use case points," in *Companion to the 20th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, 2005, pp. 257–265.

[20] W. Kurniawan, S. Sholiq, and T. Sutanto, "Penentuan Effort Rate Pada Estimasi Effort Menggunakan Metode Use Case Point Untuk Pengembangan Perangkat Lunak Website Kepemerintahan," *Jurnal JSIKA*, vol. 2, no. 2, pp. 61–71, 2013.

[21] D. Kurniadi, "Perancangan Arsitektur Sistem E-academic dengan Konsep Kampus Digital Menggunakan Unified Software Development Process (USDP)," *Jurnal Wawasan Ilmiah.*, vol. 5, no. 10, pp. 1–16, 2014.

[22] D. Kurniadi and A. Mulyani, "Implementasi Pengembangan Student Information Terminal (S-IT) Untuk Pelayanan Akademik Mahasiswa," *Jurnal Algoritma.*, vol. 13, no. 1, pp. 437–442, 2016.

Check 31. 2018 Scopus - Full - Software size measurement of student information terminal with use case point.pdf

ORIGINALITY REPORT

28%	15%	20%	17%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	D Kurniadi, A Mulyani, Y Septiana, H Aulawi. "Estimated software measurement base on use case for online admission system", IOP Conference Series: Materials Science and Engineering, 2018 Publication	7%
2	icieve.conference.upi.edu Internet Source	6%
3	Karto Iskandar, Ford Lumban Gaol, Benfano Soewito, Harco Leslie Hendric Spits Warnars, Raymondus Kosala. "Software size measurement of knowledge management portal with use case point", 2016 International Conference on Computer, Control, Informatics and its Applications (IC3INA), 2016 Publication	3%

4	Submitted to Binus University International Student Paper	2%
5	Submitted to Universitas Brawijaya Student Paper	2%
6	www.jatit.org Internet Source	2%
7	Herman Bedi Agtriadi, Natalia Chandra, Harco Leslie Hendric Spits Warnars, Ford Lumban Gaol. "Software size measurement with use case point for employee application software at STT-PLN", 2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom), 2017 Publication	1%
8	Safrizal, Harco Leslie Hendric Spits Warnars, Ford Lumban Gaol, Edi Abdurachman. "Use case point as software size measurement with study case of Academic Information System", 2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom), 2017 Publication	1%
9	www.rbmcollege.ac.in	

	Internet Source	1%
10	jurnal.amikgarut.ac.id Internet Source	1%
11	ukm.pure.elsevier.com Internet Source	1%
12	irep.iium.edu.my Internet Source	1%
13	Arman Syah Putra, Harco Leslie Hendric Spits Warnars. "Intelligent Traffic Monitoring System (ITMS) for Smart City Based on IoT Monitoring", 2018 Indonesian Association for Pattern Recognition International Conference (INAPR), 2018 Publication	1%
14	Submitted to Newham College of Further Education, London Student Paper	1%

Exclude quotes	On	Exclude matches	< 1%

Exclude bibliography On